

Remarks

Claims 1-21 are pending.

It is noted that the present Office Action is in response to the Request for Continued Examination (RCE) as filed on August 16, 2006.¹

It is not apparent from the Office Action that the Examiner has considered the Second Supplemental Information Disclosure Statement as filed on February 5, 2007. The Examiner is requested to consider the reference and make this of record.

REJECTIONS UNDER 35 U.S.C. § 103(a)

The Examiner rejects Claims 1-21 on the ground of being unpatentable over “Phishing in Alternate Data Streams” (Berghel et al.) in view of U.S. Patent No. 6,744,450 (Zimniewicz et al.).

Berghel et al. discloses primary and alternate data streams (ADSs) in the New Technology File System (NTFS) of Microsoft. This reference also discloses that (page 1) a “large number of alternate data streams (ADSs) may be associated with a single primary data stream (PDS)”, that (pages 3-4) a user renames <calc.exe> as the ADS, <d.exe>, and associates it with an empty text file <test.txt> by employing a command line of a DOS command prompt window, and that (page 6) there is “malware” that takes advantage of ADSs (e.g., W2k.stream).

Zimniewicz et al. (Abstract) discloses a system and method for a suite integration toolkit (SIT) allowing for the provision and display of a set of installation actions. The SIT is used to install a suite of applications having multiple components and sub-components. Zimniewicz et al. (col. 6, ll. 35-49) (emphasis added) states that it:

allows the addition and customization of install actions definable by application components to be installed. Further, a new user interface (UI) is presented that allows a logical and easily understandable presentation of the current state and defined actions available and selected for an installation application. This system is embodied in a Suite Integration Toolkit (SIT) and utilizes a common architecture used for a setup database file (setup.sdb) to identify components and their available actions to be performed during the installation and setup thereof. SIT complements, but does not replace, existing installer technology. SIT works as an integration layer for disparate setups but does not perform core setup tasks such as file copy, registry operations, etc., although it may be expanded to do so if desired.

¹ The Office has withdrawn the previous Office Action, mailed on November 2, 2006, which the Office improperly mailed to an incorrect correspondence address.

It is submitted that this is the only recital of the term “installer” by Zimniewicz et al.. Furthermore, Zimniewicz et al. does not recite any of the terms “Primary Data Stream file” or “Alternate Data Stream file”, much less any writing a Primary Data Stream file or an Alternate Data Stream file to a New Technology File Structure logical volume from an installer.

Claim 1 recites, *inter alia*, a method for secure installation and operation of software comprising: employing a New Technology File Structure logical volume; employing an installer; writing a Primary Data Stream file to the New Technology File Structure logical volume from the installer; associating data with the Primary Data Stream file; and writing the associated data to the New Technology File Structure logical volume as an Alternate Data Stream file from the installer.

Claim 1 recites employing an *installer*; writing a Primary Data Stream file to a New Technology File Structure logical volume *from* such *installer*; *and* writing associated data with the Primary Data Stream file and to the New Technology File Structure logical volume as an Alternate Data Stream file *from* such *installer*.

The Examiner admits (Office Action, page 4) that Berghel et al. does not disclose an installer.² As such, it is submitted that Berghel et al. does not disclose “employing an installer” as stated (Office Action, page 3) by the Examiner. Therefore, it is submitted that Berghel et al. does not disclose writing a Primary Data Stream file to a New Technology File Structure logical volume “from said installer” as stated (Office Action, page 4) by the Examiner, or writing associated data to a New Technology File Structure logical volume as an Alternate Data Stream file “from said installer” as stated (Office Action, page 4) by the Examiner.

Hence, Berghel et al. does not teach or suggest the refined recital of employing an *installer*; writing a Primary Data Stream file to a New Technology File Structure logical volume *from* such *installer*; *and* writing associated data with the Primary Data Stream file to the New Technology File Structure logical volume as an Alternate Data Stream file *from* such *installer*.

As to the term “installer”, the Examiner relies (Office Action, page 4) upon Zimniewicz et al. (“FIG. 3, 4b, 4c and related discussion”; col. 3, ll. 5-22). In connection with Figure 3, it is stated that the Setup Manager installs 90 required baseline components, and calls 92 the UI Manager to display progress information during baseline install

² It is submitted that the Examiner has abandoned the view that an “installer” is “inherent” to Berghel et al..

(preferably, progress will be displayed during silent baseline as well). The Setup Manager also installs 102, when necessary, components needed to achieve the scenario baseline if it differs from the suite baseline. This may occur, e.g., when a third party modifies a scenario data file, but neglects to modify the setup data file. As a result, the scenario baseline is no longer a subset of the suite baseline and requires the installation of additional components. The Setup Manager also launches pre-install 104, install 106, and post-install 108 loops and sends messages to components/sub components to perform actions during these installation stages before the process terminates 110. In Figure 4b, the user can 150 click Next to install 156 the scenario baseline components and continue w/installation, click Cancel 152 to exit setup 126, or click Back 154 to select a different scenario. In Figure 4c, an installation summary is provided 160, and the Setup Manager sends 166 an install message to all selected components and sub components.

Zimniewicz et al. also discloses (col. 3, ll. 4-22) that:

Unfortunately, while such programs can provide a “forklift” or recovery option that would look for the data store and only reinstall the application without requiring the user to reconfigure the machine, no such additional action of this type is allowed under the current check-box framework. Therefore, a user could only choose “reinstall all,” which action would require reconfiguration of the machine and migration of the data. Such actions typically take several days to complete for most organizations.

Therefore, there exists a need in the art to overcome these and other problems existing with the current state of suite installation programs. Specifically, there exists a need to present the installation actions to a user in a logical and easy to understand way that allows them to clearly see the state of the components affected, and what actions will be performed. Further, there exists a need to allow components to define custom installation actions and allow these custom actions to be included at the installation application run time.

Zimniewicz et al. further discloses (col. 8, ll. 45-47) that a message is written out to an error log indicating why an install cannot proceed.

Therefore, Zimniewicz et al., which does not disclose any Primary Data Stream file or any Alternate Data Stream file, adds nothing to Berghel et al. regarding writing a Primary Data Stream file or an Alternate Data Stream file to a New Technology File Structure logical volume from an installer.

On page 4 of the Office Action, the Examiner concludes that it would have been obvious “to incorporate the method of having an installer to install the application as

taught by Zimniewicz into the method of installation of ADS PDS files as taught by Berghel.” However, it is respectfully submitted that this conclusion is clearly in error since it has been shown, above, that there is no “employing an installer” much less any writing a Primary Data Stream file to a New Technology File Structure logical volume “from an installer” or any writing associated data to a New Technology File Structure logical volume as an Alternate Data Stream file “from an installer” in Berghel et al.. Although Zimniewicz et al. discloses a suite integration toolkit (SIT) to install a suite of applications having multiple components and sub-components, it does not teach or suggest any writing a Primary Data Stream file to a New Technology File Structure logical volume from an installer or any writing associated data to a New Technology File Structure logical volume as an Alternate Data Stream file from an installer. At best, the hypothetical combination of Berghel et al. and Zimniewicz et al. would provide an installation of a “suite of applications having multiple components and sub-components”. This does not teach or suggest the refined recital of Claim 1.

Accordingly, for the above reasons, Claim 1 patentably distinguishes over the references.

Claims 2-10 depend either directly or indirectly from Claim 1 and patentably distinguish over the references for at least the same reasons.

Furthermore, Claim 5 recites creating a Primary Data Stream directory chain; writing the Primary Data Stream directory chain to the New Technology File Structure logical volume *from* the *installer*; writing the Primary Data Stream file to the Primary Data Stream directory chain in the New Technology File Structure logical volume *from* the *installer*; associating the data with the Primary Data Stream directory chain or the Primary Data Stream file by creating and closing the Alternate Data Stream file; and *installing* the associated data to the New Technology File Structure logical volume as the Alternate Data Stream file *from* the *installer*.

The Examiner’s reliance upon Berghel et al. for “from said installer” has been dealt with above in connection with Claim 1.

Since the references neither teach nor suggest the refined recital of Claim 1, they clearly neither teach nor suggest these additional limitations which further patentably distinguish over the references.

Furthermore, Claim 6 recites employing an *installation* file comprising the Primary Data Stream file, the Alternate Data Stream file, installation instructions, the Primary Data Stream directory chain, and an End User License Agreement.

It is respectfully submitted that a Windows directory is not an installation file as was stated by the Examiner on page 8 of the Office action.

Claim 6 depends directly from Claim 5 and indirectly from Claim 1 and includes all of the limitations of those claims. Since the references neither teach nor suggest the refined recital of Claim 5, they clearly neither teach nor suggest these additional limitations which further patentably distinguish over the references.

Furthermore, Claim 8 recites employing as the associated data first data; employing as the Alternate Data Stream file a first Alternate Data Stream file; employing second data; associating the second data with the Primary Data Stream file; and writing the associated second data to the New Technology File Structure logical volume as a second Alternate Data Stream file *from the installer*.

The Examiner's reliance upon Berghel et al. for "from said installer" has been dealt with above in connection with Claim 1.

Since the references neither teach nor suggest the refined recital of Claim 1, they clearly neither teach nor suggest these additional limitations which further patentably distinguish over the references.

Furthermore, Claim 10, which depends from Claim 1 and includes all of the limitations thereof, recites employing an *installation* file; defining in the installation file a Primary Data Stream directory chain, the Primary Data Stream file, the Alternate Data Stream file, and at least one information file; displaying the at least one information file from the installation file; creating the Primary Data Stream directory chain in the New Technology File Structure logical volume; copying the Primary Data Stream file from the installation file to the Primary Data Stream directory chain in the New Technology File Structure logical volume; and copying the Alternate Data Stream file from the installation file to the Primary Data Stream directory chain in the New Technology File Structure logical volume.

It is respectfully submitted that a Windows directory is not an installation file as was stated by the Examiner on page 8 of the Office action.

Since the references neither teach nor suggest the refined recital of Claim 1, they clearly neither teach nor suggest these additional limitations which further distinguish over the references.

Claim 11 is an independent claim which recites, *inter alia*, a computer system for secure installation and operation of software comprising: a processor; a first drive adapted for access by the processor; a second drive adapted for access by the processor, the second drive including a New Technology File Structure logical volume; and an installer operatively

associated with the first drive, the installer cooperating with the processor to write a Primary Data Stream file to the New Technology File Structure logical volume, associate data with the Primary Data Stream file, and write the associated data to the New Technology File Structure logical volume as an Alternate Data Stream file.

The Examiner rejects Claim 11 “under the same rationale” set forth in connection with the rejection of Claim 1. The Examiner admits (Office Action, page 4) that Berghel et al. does not disclose an installer. As such, it is submitted that Berghel et al. does not disclose an installer operatively associated with a first drive, an installer cooperating with a processor to write a Primary Data Stream file to a New Technology File Structure logical volume, and to write associated data to a New Technology File Structure logical volume as an Alternate Data Stream file.

As to the term “installer”, the Examiner relies (Office Action, page 4) upon Zimniewicz et al. (“FIG. 3, 4b, 4c and related discussion”; col. 3, ll. 5-22). As has been shown, above, in connection with Claim 1, Zimniewicz et al., which does not disclose any Primary Data Stream file or any Alternate Data Stream file, adds nothing to Berghel et al. regarding writing a Primary Data Stream file or an Alternate Data Stream file to a New Technology File Structure logical volume from an installer.

On page 4 of the Office Action, the Examiner concludes that it would have been obvious “to incorporate the method of having an installer to install the application as taught by Zimniewicz into the method of installation of ADS PDS files as taught by Berghel.” However, it is respectfully submitted that this conclusion is in error since it has been shown, above, that there is no “installer operatively associated with a first drive” much less any installer cooperating with a processor to write a Primary Data Stream file to a New Technology File Structure logical volume, and to write associated data to a New Technology File Structure logical volume as an Alternate Data Stream file in Berghel et al.. Although Zimniewicz et al. discloses a suite integration toolkit (SIT) to install a suite of applications having multiple components and sub-components, it does not teach or suggest any installer cooperating with a processor to write a Primary Data Stream file to a New Technology File Structure logical volume, and to write associated data to a New Technology File Structure logical volume as an Alternate Data Stream file. At best, the hypothetical combination of Berghel et al. and Zimniewicz et al. would provide an installation of a “suite of applications having multiple components and sub-components”. This does not teach or suggest the refined recital of Claim 11.

The references do not teach or suggest the refined recital of a computer system for secure installation and operation of software comprising: an *installer* operatively associated with a first drive, such *installer* cooperating with a processor to write a Primary Data Stream file to a New Technology File Structure logical volume, associate data with such Primary Data Stream file, and write such associated data to such New Technology File Structure logical volume as an Alternate Data Stream file.

Therefore, for the above reasons, Claim 11 patentably distinguishes over the references.

Claims 12-20 depend either directly or indirectly from Claim 11 and patentably distinguish over the references for at least the same reasons.

Furthermore, Claim 12 recites that the New Technology File Structure logical volume includes a directory chain or a system directory; and that the *installer* installs the Primary Data Stream file in the directory chain or the system directory of the New Technology File Structure logical volume.

The Examiner's reliance upon Berghel et al. for an "installer" has been dealt with above in connection with Claim 1.

Since the references neither teach nor suggest the refined recital of Claim 11, they clearly neither teach nor suggest these additional limitations which further patentably distinguish over the references.

Furthermore, Claim 16 recites that the *installer* cooperates with the processor to create a Primary Data Stream directory chain, to write the Primary Data Stream directory chain to the New Technology File Structure logical volume, to write the Primary Data Stream file to the Primary Data Stream directory chain in the New Technology File Structure logical volume, to associate the data with the Primary Data Stream directory chain or the Primary Data Stream file, and to install the associated data to the New Technology File Structure logical volume as the Alternate Data Stream file.

The Examiner's reliance upon Berghel et al. for an "installer" has been dealt with above in connection with Claim 1.

Since the references neither teach nor suggest the refined recital of Claim 11, they clearly neither teach nor suggest these additional limitations which further patentably distinguish over the references.

Furthermore, Claim 17 recites that the *installer* comprises an *installation* file comprising the Primary Data Stream file, the Alternate Data Stream file, installation instructions, a Primary Data Stream directory chain, and an End User License Agreement.

Claim 17 further patentably distinguishes over the references for similar reasons as were discussed above in connection with Claim 6.

Furthermore, Claim 18 recites that the processor includes a display; and that the *installer* cooperates with the processor to display the installation instructions and the End User License Agreement on the display.

The Examiner's reliance upon Berghel et al. for an "installer" has been dealt with above in connection with Claim 1.

Since the references neither teach nor suggest the refined recital of Claims 11 and 17, they clearly neither teach nor suggest these additional limitations which further patentably distinguish over the references.

Furthermore, Claim 20 recites that the processor includes a display; that the *installer* comprises an *installation* file including a Primary Data Stream directory chain, the Primary Data Stream file, the Alternate Data Stream file, and at least one information file; and that the *installer* cooperates with the processor to display the at least one information file from the installation file to the display, to create the Primary Data Stream directory chain in the New Technology File Structure logical volume, to copy the Primary Data Stream file from the installation file to the Primary Data Stream directory chain in the New Technology File Structure logical volume, and to copy the Alternate Data Stream file from the installation file to the Primary Data Stream directory chain in the New Technology File Structure logical volume.

It is respectfully submitted that a Windows directory is not an installation file as was stated by the Examiner on page 8 of the Office action.

Claim 20 further patentably distinguishes over the references for similar reasons as were discussed above in connection with Claim 10.

Claim 21 is an independent method claim which recites, *inter alia*, a method for secure installation and operation of software comprising: employing a computer-readable medium including a New Technology File Structure logical volume; employing an installer; writing a Primary Data Stream file to the New Technology File Structure logical volume of the computer-readable medium from the installer; associating data with the Primary Data Stream file; and writing the associated data to the New Technology File Structure logical volume of the computer-readable medium as an Alternate Data Stream file from the installer.

The Examiner rejects Claim 21 "under the same rationale" set forth in connection with the rejection of Claim 1. The Examiner admits (Office Action, page 4) that Berghel et al. does not disclose an installer. As such, it is submitted that Berghel et al. does

not disclose employing an installer; writing a Primary Data Stream file to a New Technology File Structure logical volume of a computer-readable medium from an installer; and writing associated data to a New Technology File Structure logical volume of a computer-readable medium as an Alternate Data Stream file from an installer.

As to the term “installer”, the Examiner relies (Office Action, page 4) upon Zimniewicz et al. (“FIG. 3, 4b, 4c and related discussion”; col. 3, ll. 5-22). As has been shown, above, in connection with Claim 1, Zimniewicz et al., which does not disclose any Primary Data Stream file or any Alternate Data Stream file, adds nothing to Berghel et al. regarding writing a Primary Data Stream file to a New Technology File Structure logical volume of a computer-readable medium from an installer; and writing associated data to a New Technology File Structure logical volume of a computer-readable medium as an Alternate Data Stream file from an installer.

On page 4 of the Office Action, the Examiner concludes that it would have been obvious “to incorporate the method of having an installer to install the application as taught by Zimniewicz into the method of installation of ADS PDS files as taught by Berghel.” However, it is respectfully submitted that this conclusion is in error since it has been shown, above, that there is no “employing an installer” much less any writing a Primary Data Stream file to a New Technology File Structure logical volume of a computer-readable medium from an installer; and writing associated data to a New Technology File Structure logical volume of a computer-readable medium as an Alternate Data Stream file from an installer in Berghel et al.. Although Zimniewicz et al. discloses a suite integration toolkit (SIT) to install a suite of applications having multiple components and sub-components, it does not teach or suggest any writing a Primary Data Stream file to a New Technology File Structure logical volume of a computer-readable medium from an installer; or any writing associated data to a New Technology File Structure logical volume of a computer-readable medium as an Alternate Data Stream file from an installer. At best, the hypothetical combination of Berghel et al. and Zimniewicz et al. would provide an installation of a “suite of applications having multiple components and sub-components”. This does not teach or suggest the refined recital of Claim 21.

For similar reasons as were discussed above in connection with Claim 1, the references do not teach or suggest employing an *installer*; writing a Primary Data Stream file to a New Technology File Structure logical volume of a computer-readable medium *from* such *installer*; associating data with such Primary Data Stream file; and writing such

associated data to such New Technology File Structure logical volume of such computer-readable medium as an Alternate Data Stream file *from* such *installer*.

Therefore, for the above reasons, Claim 21 patentably distinguishes over the references.

Reconsideration and early allowance are respectfully requested.

Respectfully submitted,

/Kirk D. Houser/

Kirk D. Houser
Registration No. 37,357
Attorney for Applicant

(412) 566-6083